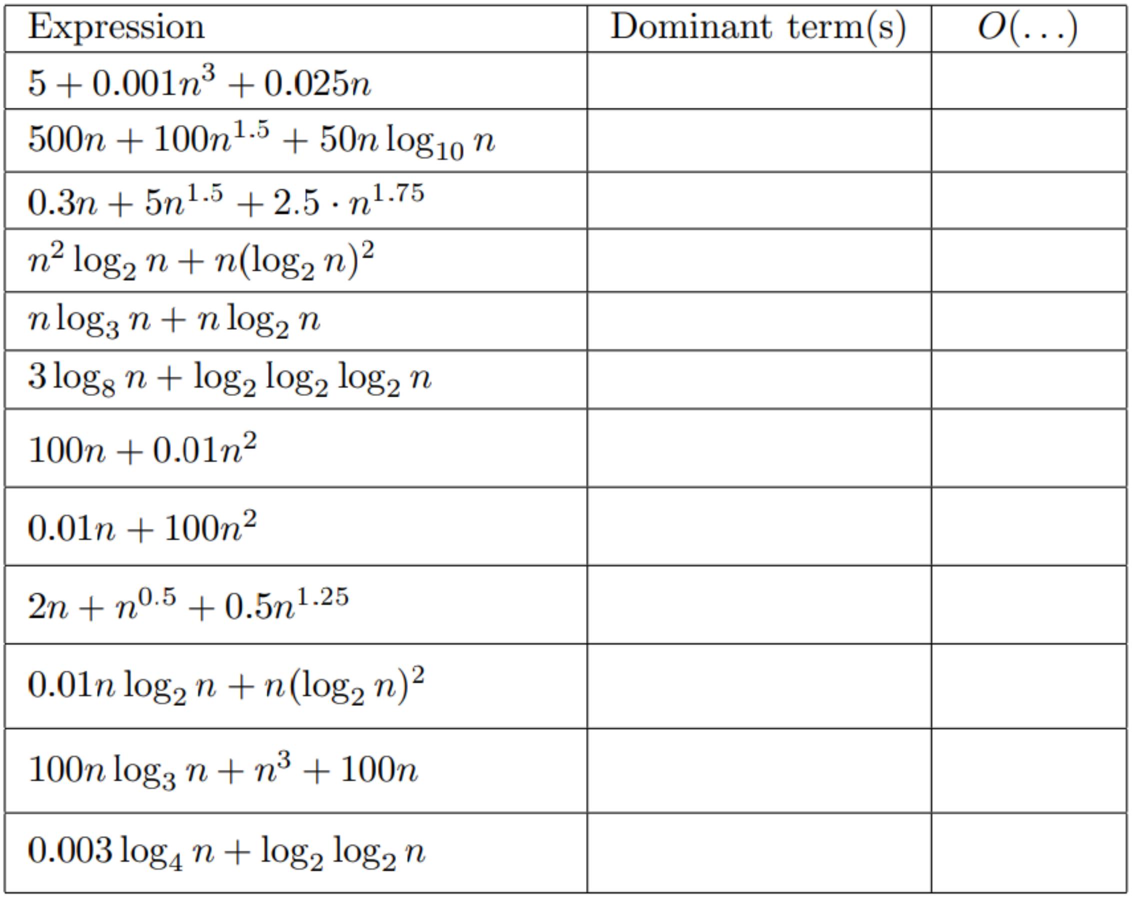
Loyola University Chicago
Department of Computer Science
COMP 272/400C: Data Structures II (Spring 2023)

Prof: Yas Silva

Analysis of Algorithms

A. [**]Assume that each of the expressions below gives the processing time T(n) spent by an algorithm for solving a problem of size n. Select the dominant term(s) having the steepest increase in n and specify the lowest Big-O complexity of each algorithm.



Expression	Dominant term(s)	$O(\ldots)$
$5 + 0.001n^3 + 0.025n$		
$500n + 100n^{1.5} + 50n\log_{10} n$		
$0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$		
$n^2 \log_2 n + n(\log_2 n)^2$		
$n \log_3 n + n \log_2 n$		
$3\log_8 n + \log_2 \log_2 \log_2 n$		
$100n + 0.01n^2$		
$0.01n + 100n^2$		
$2n + n^{0.5} + 0.5n^{1.25}$		
$0.01n \log_2 n + n(\log_2 n)^2$		
$100n \log_3 n + n^3 + 100n$		
$0.003 \log_4 n + \log_2 \log_2 n$		

B. What is the time complexity of the following algorithms? Find the expression that estimates the number of operations in terms of the input size (n) and express it using the Big-O notation.

1. [*]

```
void method1(int n){
   int a = 0, b = 0;
   for (i = 0; i < n; i++)
       a = a + 1;
   for (j = 0; j < n; j++)
       b = b + 2;
}
```

2. [*]

```
void method2(int n){
   int a = 0, b = 0;
   for (i = 0; i < n; i++) {
       a = a + 1;
       for (j = 0; j < n; j++)
           b = b + 2;
   }
}
```

3. [**]

```
void method3(int n){
   int a = 0, b = 0;
   for (i = n; i > 0; i=i/2)
       a = a + 1;
}
```

4. [***]

```
void method4(int n){
   int a = 0, b = 0;
   for (i = 0; i < n; i++) {
      for (j = 0; j < i; j++)
           b = b + 2;
   }
}
```

5. [🔥]

```
int method5(int n){
  int count = 0;
  for (int i = n; i > 0; i /= 2)
     for (int j = 0; j < i; j++)
        count += 1;
  return count;
}
```

$$\sum_{k=0}^{n} z^k = \frac{1 - z^{n+1}}{1 - z}$$

(a) O(Logn)
(b) O(nLognLogn)
(c) O(n)
(d) O(nLogn)
(e) O(n^2)

C. [**] Sort the following functions in increasing order of asymptotic complexity.

f1(n) = nLogn
f2(n) = n^(3/2)
f3(n) = 2^n
f4(n) = Logn
f5(n) = n^(Logn)

D. [***] What is the time complexity (using Big O notation) of the following two methods?

```
void fun1(int n, int arr[])
{
    int i = 0, j = 0;
    for(; i < n; ++i)
    {
        j = 0;
        while(j < n && arr[i] < arr[j])
            j++;
    }
}

void fun2(int n, int arr[])
{
    int i = 0, j = 0;
    for(; i < n; ++i)
        while(j < n && arr[i] < arr[j])
            j++;
}
```

E. [**] Which of these expressions is not O(n^2) using the formal defintion?

(A) (10^20) * n
(B) n^1.99
(C) (2^10) * n + 12888
(D) n^3 / (sqrt(n))
(E) nLogn